NATIONAL MARINE FISHERIES SERVICE
SOUTHWEST FISHERIES CENTER
P.O. BOX 271
LA JOLLA, CA 92038

NOVEMBER 1988

CONTOUR MAPPING AND THE CALCULATION OF AREAS BETWEEN 10M DEPTH CONTOURS ALONG THE COASTS OF CALIFORNIA, OREGON AND WASHINGTON

by

Karin Forney

ADMINISTRATIVE REPORT LJ-88-23

This Administrative Report is issued as an informal document to ensure prompt dissemination of preliminary results, interm reports and special studies. We recommend that it not be abstracted or cited.

Contour mapping and the calculation of areas between 10m depth contours along the coasts of California, Oregon and Washington.

Karin Forney
Southwest Fisheries Center
National Marine Fisheries Service
P.O. Box 271
La Jolla, CA 92038

October 31, 1988

Administrative Report LJ-88-23

CONTENTS

Page

ABSTRACT 1

INTRODUCTION. 1

MATERIALS AND METHODS 2

PROBLEMS 5

RECOMMENDATIONS 8

ACKNOWLEDGEMENTS 8

LITERATURE CITED 9

APPENDIX A 13

APPENDIX B 16

APPENDIX C 18

LIST OF TABLES AND FIGURES

		Page
TABLE 1.	Data file information. Coordinates refer to the southeast corner of the 1° square.	10
TABLE 2.	Areas between contour lines for eight regions along the Pacific U.S. coast (in square nautical miles).	11
FIGURE 1.	Geographic regions used for area calculations.	12

Contour mapping and the calculation of areas between 10m depth contours along the coasts of California, Oregon and Washington.

Karin Forney
Southwest Fisheries Center
National Marine Fisheries Service
P.O. Box 271
La Jolla, Ca 92038.

ABSTRACT

Fine-scale contours (10m increments) are generated for coastal waters (0-150m) of California, Oregon and Washington from data files containing digitized depth information. Areas between each of the contour levels are calculated, and problems encountered are discussed in detail. The results are used by Barlow (1988) to estimate harbor porpoise abundance.

INTRODUCTION

The availability of contour maps for coastal waters is largely limited to contour intervals of 10 fathoms or larger. Although this is sufficient for most purposes, some studies may require a more detailed breakdown of depth contours. For example, harbor porpoise (Phocoena phocoena) off the coast of California, Oregon and Washington occur mainly in shallow waters of less than 150m depth. Barlow (1988) proposes three models of harbor porpoise distribution as a function of depth, and thus a finer subdivision of contours is desirable. The determination of areas between depth contours in 10m increments, described below, has enabled Barlow to adjust his estimate of harbor porpoise density according to his three proposed models, and produce a more accurate estimate of total harbor porpoise abundance. The procedures described below are applicable to other situations in which more specific contours are desired, and hopefully will allow others to avoid the problems that were encountered in generating depth contours for harbor porpoise abundance estimation.

MATERIALS AND METHODS

The intitial data base, a subset of the complete hydrographic data base maintained by the National Ocean Service (NOS), was purchased by the Southwest Fisheries Center for contouring purposes. The data were provided in files representing single squares of 1° latitude and longitude. Within these files, most lines contained coordinates (degrees, minutes and seconds of latitude and longitude), a depth measurement, and a code specifying the format of the depth measurement (nine different possibilities involving fathoms/feet/meters with feet/tenths/fractions). In addition, there were a number of lines with other information (e.g. bottom characteristics at certain points), which were also identified by code. The files obtained are listed in Table 1, along with the number of data points they contained.

These raw data files were loaded onto the CRAY X-MP, located at the San Diego Supercomputer Center (hereafter referred to as the CRAY), from magnetic tapes and converted to CTSS format. A program written in CRAY FORTRAN transformed them into data files containing only depth information in the format LONGITUDE, LATITUDE, DEPTH (IN METERS). This program is listed in Appendix A. Furthermore, some of the areas, particularly along the coast of Oregon and Washington, contained too few data points to create a contour map, or contained points which only covered a portion of the entire 1° square, so navigation charts were used to digitize the coordinates of additional depth points into the data base. A GP-7 Graphbar Model sonic digitizer (Science Accessories Corporation) was utilized, and a program was written in Microsoft QuickBASIC for the IBM PC to process the digitized information (listed in Appendix B).

Once the data files were complete, contour maps were created in two steps. First, since the data points were irregularly distributed, they had to be transformed into a grid with a single depth point for each grid square. Dr. Bob Parker of Scripps Institute of Oceanography provided a program (ZGRID) to do this. This program had been written in VAX FORTRAN, and therefore needed to be converted into CRAY FORTRAN. The intitial plan was to further modify this program to allow for a 240x240 grid, with each grid square representing a 15-second square of actual latitude and longitude. However, preliminary runs of ZGRID using its maximum resolution of 100x100 produced sufficiently detailed contours, so that it was not necessary to change grid size. However, some unanticipated problems with ZGRID were revealed, and other modifications became necessary (see PROBLEMS below).

The second step of contouring involved use of a program available in the plotting libraries of the CRAY, called CONTOUR. Two files must be present before CONTOUR can be used: one containing the grid of depth measurements and one containing instructions and information for the contour plot. The first file is created by running ZGRID with the files containing irregularly distributed data points. The format of the second file is described in depth by the on-line documentation available for CONTOUR on the CRAY. An example of such a file is listed in Appendix C.

In addition to the problems with ZGRID mentioned above, the preliminary runs of CONTOUR revealed further problems, in particular with the coastline. After some trial and error in resolving this problem (see PROBLEMS below), it was determined that the coastline could not be accurately placed because there were no reference points beyond the zero depth contour (i.e. elevation points on land). To correct for this, I digitized a hypothetical -20 meter contour (i.e. 20 meters elevation) from navigation charts and appended this to the data base. I positioned this hypothetical contour at roughly the same distance from the coast as the 10 fathom (=18m) depth contour. This hypothetical elevation contour was chosen rather than actual elevation points for reasons described below. The resulting coastline contour corresponded extremely well to the coastline on the navigation charts.

Once satisfactory contour plots were produced for all regions along the coast, the next step was to calculate areas between each of the contour levels in the eight regions previously defined by Barlow (1988, see Figure 1). To accomplish this, the program CONTOUR was modified to output the actual coordinates of each of the contour lines in lieu of plotting the coordinates. This task was completed with the help of SDSC consultant Tom Metzger.

Unfortunately, the only way to output the coordinates was in the sequence in which the program determined them, and this was not the format required for the area calculation program (see below). The only sorting that was possible at this point was to determine which of the eight regions a set of coordinates fell into, and to write them to one of eight different files. Within each region, this left the contour coordinates organized primarily by 1° square in ascending order of contour depth, alternately running north and south. To calculate areas, it was necessary to convert these files into files of continuous contour segments within a region (spanning several 1° squares), uniformly running in a single direction.

The extensive editing necessary was done on the IBM PC after the files had been downloaded from the CRAY. First, a program was written in Microsoft QuickBASIC to separate the coordinates according to the contour level they represented. Thus each regional file was divided into 16 separate files, each containing only coordinates for a single depth contour (0-150m). These individual files then had to be examined and edited to create continuous contour lines within the regions.

To assist in detecting all items requiring editing, I manually pasted together continuous contour maps of the eight regions, which showed all lines exactly as they were represented in the data files. It was frequently necessary to correct coordinates to obtain a smooth transition where the contour lines from two different 1° squares connected. Additionally, any 'islands' of a given contour depth had to be moved into separate files, and some remaining errors caused by false points in the data base (see PROBLEMS below) had to be removed. The files were also re-organized so that all contour coordinates went from south to north, except for the coastline contour, which was to be used as a reference line for all contour levels, and therefore went from north to south. VEDIT PLUS proved to be a suitable editing package for this purpose, as it allows multiple files to be open at the same time. The north-south inversion of portions of the files was accomplished with a simple program written in Microsoft QuickBASIC.

When the editing was completed, these new data files were uploaded to the CRAY for final processing. A FORTRAN subroutine used to calculate areas on a sphere (SPHAREA, written by Joseph Joy at the Southwest Fisheries Center), was obtained and converted to CRAY FORTRAN, and a main program was written from which it could be called. The subroutine was additionally modified to allow for seconds of latitude and longitude in the coordinates, as it initially only allowed for degrees and minutes. Each of the individual contour files was then combined with the coastline file, and the first point in the file was repeated at the end to provide a closed area.

Initially, many of the files were rejected by the area calculation program, because there were duplicate data points, presumedly due to the repeated conversion of coordinates into different formats and scales. The error message provided by SPHAREA gave line numbers for the identical entries in the file, so it was relatively easy to eliminate the excess points. After these points had been eliminated, the program quickly produced areas for each of the files. Some of the files representing small 'islands' actually produced an area of zero, because the areas were rounded to the nearest integer. These islands were later not included in the calculation of regional totals.

Once all areas had been obtained with the program, the total area for each of the contour levels, A, was calculated according to the following rules:

$$A_i = a_i - a_{i-1} + \alpha_i - \beta_i$$

where a_i = area between coast and contour level i
a_{i-1} = area between coast and contour level i-1
α_i = area of any islands of contour level i
β_i = area of any islands of a different contour level that fall within the area of contour level i.

The total areas calculated for each contour level in the eight regions are summarized in Table 2.

PROBLEMS

Throughout the project, there were a number of unanticipated problems and complications. The most critical ones are outlined below along with the solutions found.

To begin with, the NOS data files were not optimal for the purpose of fine-scale contouring, because the data points are distributed highly irregularly, and are clumped in areas of interest to specific agencies. For example, although the file for the 1° square with the San Francisico Bay area contains an extremely large number of data points (> 600,000, see Table 1), most of the points are inside the San Francisco Bay, and there are insufficient points outside of the bay to accurately contour the region. It was necessary to digitize additional points into the already enormous file, and most of the given points were not of any use. It later became apparent that using only the points I digitized myself would have produced good contours without the cost of processing the large NOS file. The files which contained mainly points I digitized were considerably smaller, with numbers of points in the thousands, rather than tens or hundreds of thousands.

It must be cautioned, however, that the individual depth points which I digitized are not entirely accurate, because the depth labels on the charts spanned several millimeters and varied in length. The coordinates at the start of the depth label were not the same as at the end of the label. When digitizing, I centered the stylus along the bottom edge of the depth label, and the resulting contours appeared to be accurate, but is not certain to

what degree using the left or right side of the label would have changed the contours. My guess is that it would make little difference, as long as consistency is maintained.

Initially, I had familiarized myself with a contouring package developed by the National Center for Atmospheric Research (NCAR), called NCARGRAPHICS, which is available on the CRAY. It has the useful feature of being able to create contours directly from a set of irregularly (i.e. not gridded) data points, and thus seemed to be appropriate for use with the NOS hydrographic data. After learning to use the package and testing it with sample files containing roughly two thousand data points, I attempted to test it with a medium sized file (roughly 20,000 points). However, after using over 15 minutes of CRAY CPU time, the program had still not finished, and I decided to find an alternate method which could better deal with the large number of data points. In retrospect, the best solution may have been to use only points I digitized myself, because the files would have contained no more than about 3,000 points. However, at the time I decided to switch to the programs ZGRID (which produces a grid from irregularly distributed data points), and CONTOUR (a contouring program which produces contours from a grid of points).

For the most part, this approach worked out well, but there were a few problems. The most critical problem was again related to the irregular distribution of data points. Due to a lack of points in certain areas of the 1° square, ZGRID was sometimes not able to assign a depth to all grid points, and therefore these points received a very large value (10^{35}). Presumably, these points should have been ignored by the contouring routine, but for some reason they were not. An attempt was made to correct this problem in the program CONTOUR, but due to the size and complexity of CONTOUR, I was not able to successfully do so. However, I was able to get around this problem by determining the areas with the large values and eliminating them from the gridding process by not specifying the full 1° square. It must be pointed out that this only worked because the areas were confined to the edges of the 1° square; this problem would probably be more difficult to resolve if the areas with missing points were more centrally located.

The most time consuming problem encountered was the inability of the contouring package to accurately place a coastline contour. The reason for this was not initially apparent, and only through trial and error was I able to resolve the problem. Initially, I thought that it was due to a lack of points along the coast, so I obtained additional coastline points from available data files, and added them to the original data files. However, this did not resolve the problem and the coastline still appeared distorted.

My next guess was that the process of going from contour to grid and back to contour was distorting the actual location of the coastline, so I made new contour plots containing only the 10-150m contours, and made additional plots of only the coastline directly from the coastal data files. This seemed to work well for most of the areas, but in the steeper coastal regions (2-4), the coast obviously did not fit well and actually crossed some of the other contours. Close examination of the coastal points in the data files for the affected regions revealed inaccuracies in the coordinates, so I eliminated these points and obtained a more accurate coastline contour by digitizing points from navigation charts. However, although the coastal points were now very accurate, the coastline still did not fit well and the problem of crossing other contour levels persisted.

At this point, I realized that the likely cause of these problems was the lack of inland reference points for determination of the zero contour. To resolve this problem, I decided to once again digitize points from charts. However, the scale of available topographic maps for the affected regions was far too detailed and I feared that the erratic contours caused by erosion of the land would again distort the coastline. The navigation charts were also not directly useable because they did not contain appropriate inland contour lines. I therefore decided to 'trick' the contouring routine with a hypothetical contour line that closely paralleled the actual coastline. Using the navigation charts, I digitized a hypothetical -20 meter contour line at roughly the same distance from the coast as the 10 fathom (=18m) depth contour. This provided a point of reference without causing distortion of the coast in either direction. Fortunately, this approach worked very well, producing coastline contours that closely resembled those on the navigation charts. Due to time constraints, these additional points were only provided for the regions where the coastline did not already provide a good fit (regions 2-4).

A further problem encountered involved the transfer of data between the CRAY and the IBM PC. I did most of the transferring overnight, because of the time required to transfer these very large files. However, on two occasions the transmission was not completed because of the time-sharing process used on the CRAY. In both instances, there presumably were large batch files running on the CRAY, which caused my transmission to be put on hold for a period exceeding the time acceptable by the communications software for the PC (MS Kermit). The two obvious precautions that are therefore necessary are to avoid busy times on the time-sharing computer, and to set up the receiving communications package so that it does not abort the file transfer as quickly if the transmisison is put on hold.

A final problem which occurred throughout the contouring process was the presence of false data points. These points usually appeared as deep holes or steep peaks in unusual places on the contour map. Some of the data points were merely entered incorrectly when digitizing, and therefore were easily found and corrected. However, there were also errors in the data files provided by NOS, which were much larger and more difficult to edit. In these cases, I found it easier to remove the incorrect contour lines produced by the false points during the final editing of the files containing the contour coordinates.

RECOMMENDATIONS

The most important recommendation I have to make to others interested in doing something similar is to use points digitized from navigation charts, rather than points received from other sources. The main benefits are that the points are then more equally distributed and the files are smaller and more manageable. With these smaller files it would then be possible to use professional contouring packages, such as NCARGRAPHICS, which generally have more flexibility and fewer problems than programs from other sources. However, I would like to emphasize that the programs ZGRID and CONTOUR were very suitable and easy to use, and allowed sufficient flexibility. The only problems I had involved the fact that they had not been thoroughly tested for a variety of contouring cases.

In addition, I would recommend adding a hypothetical elevation contour to any data files which contain a coastline, as the problem is not likely to be specific to the contouring software I used. This additional digitizing took very little time and improved the quality of the coastline contours immensely. Also, it is important to make sure that each square to be contoured does not have any substantial 'holes' of data points, to avoid problems such as were caused by the 10^{35} value assigned to missing values by ZGRID. If the points are digitized from charts, making sure there are no large holes should not be a problem, as contour lines and depth points are available in sufficiently small increments.

ACKNOWLEDGEMENTS

Jay Barlow provided the intitial guidelines for this project and assisted me throughout my work. I thank Dr. Bob Parker and Joseph Joy for making their programs available. Tom Metzger, a consultant at the San Diego Supercomputer Center, provided valuable assistance for my programming work on the CRAY. I also thank Larry Eber for providing the NOS files and for his insights on contouring.

LITERATURE CITED

Barlow, J. 1988. Harbor Porpoise (Phocoena phocoena) Abundance Estimation for California, Oregon and Washington: I. Ship Surveys. Fish. Bull. 86:417-432.

Table 1
Data file information. Coordinates refer to the southeast corner of the 1° square.

1° square Lat	Long	NOS filename (from tape)	# data points (original file)	# data points (final file)	# points digitized offshore	inland
34	120	w05263.2	141,175	131,727	0	0
35	120	w05263.5	95,088	90,814	0	0
35	121	w05263.6	65,397	58,724	0	0
36	121	w05263.9	89,021	80,498	0	86
36	122	w05263.10	23,442	21,204	0	0
37	122	w05263.13 w10188.1	699,702	683,585	1,939	79
37	123	w10188.2	10,410	10,896	516	0
38	122	(not used)	0	195	195	0
38	123	w10188.5	68,063	64,896	1,536	84
39	123	w10188.7	21,521	20,822	1,249	99
39	124	w10188.8	9,090	8,914	283	0
40	124	w06720.1	106,309	99,389	0	73
41	124	w06720.3	6,757	8,365	1,700	0
42	124	w06720.6	120	1,921	1,801	0
43	124	w06720.7	3,702	5,429	1,727	0
44	124	w06720.9	25,706	27,607	2,588	0
45	123	w06720.12	29,221	29,265	368	0
45	124	w06720.13	51	1,741	1,693	0
46	123	w06720.14	351,209	337,550	78	0
46	124	w06720.16	79,784	80,557	2,520	0
47	124	w08557.3	50,527	50,101	1,417	0
47	125	w08557.4	3,509	3,471	0	0
48	124	w11189.4	51,678	49,246	0	0
48	125	w11189.5	25,433	25,352	0	0

TABLE 2
Areas between contour lines for eight regions along the Pacific U.S. coast (in square nautical miles).

Depth range (in meters)	R 1	E 2	G 3	I 4	O 5	N 6	7	8
0-10	53	34	32	51	89	58	54	236
10-20	60	53	72	49	129	50	65	196
20-30	81	53	81	52	129	50	61	212
30-40	74	50	95	54	135	60	72	273
40-50	65	48	98	51	124	79	73	225
50-60	63	57	163	50	123	112	92	184
60-70	62	77	117	54	120	125	103	187
70-80	62	83	125	72	123	181	129	200
80-90	60	105	108	90	122	181	146	167
90-100	66	119	111	108	126	231	112	178
100-110	52	90	114	120	101	228	113	184
110-120	38	69	75	116	85	265	117	222
120-130	27	45	42	132	73	187	127	191
130-140	22	22	26	113	69	133	206	214
140-150	20	15	19	100	60	96	157	200
Totals	805	920	1278	1212	1608	2036	1627	3069

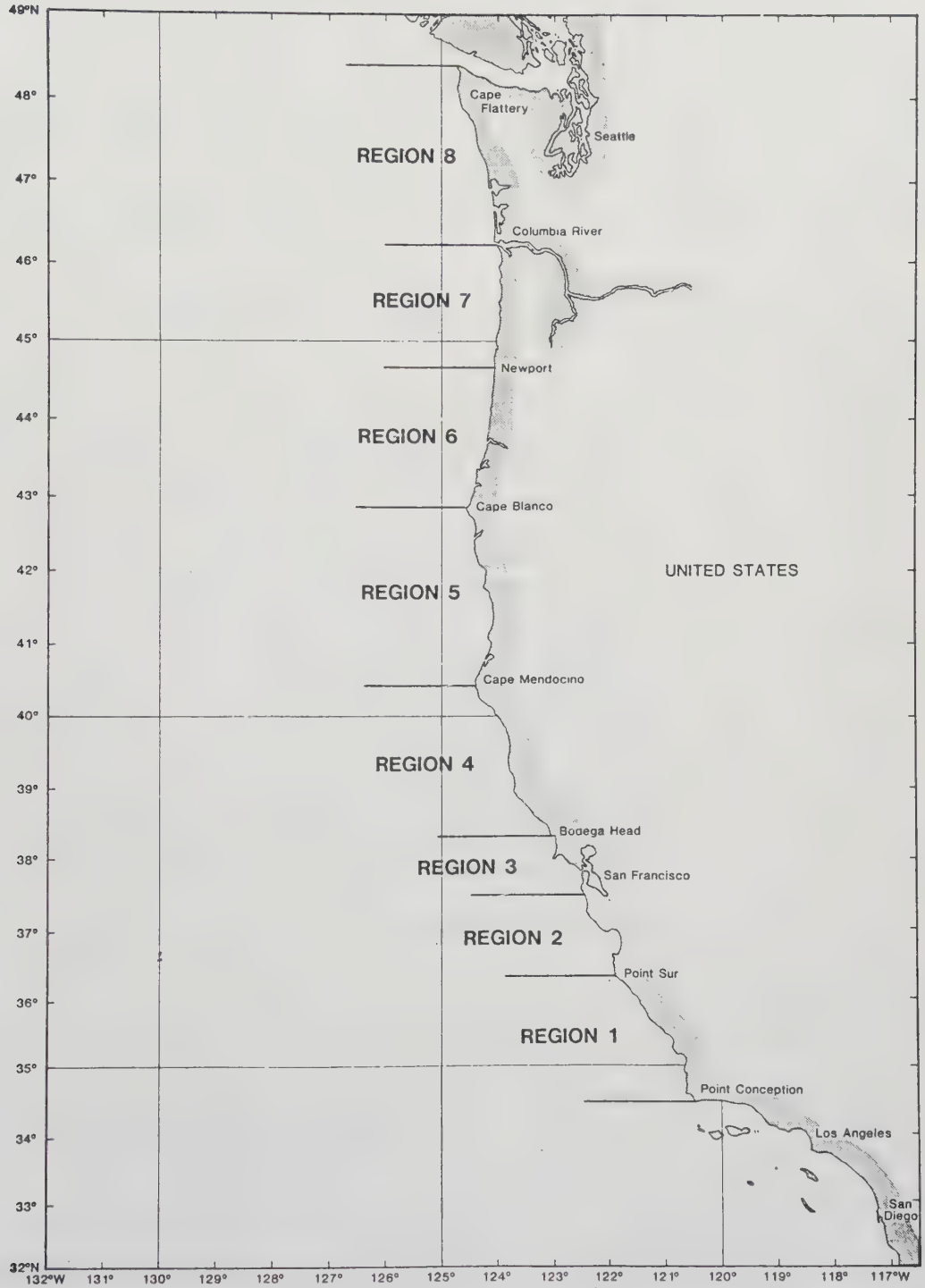

FIGURE 1. Geographic regions used for area calculations.

APPENDIX A
Program used to convert NOS hydrological data into the format required for the creation of contours.

```
C============================== CONVERT ================================
C  This program will read in a data file from the NOS Hydrographic dat
C base,  and convert the depth data into x,y,  and z values, where x i
C latitude (in degrees; 6 decimal positions), y is the longitude (same
C and z is the depth converted to meters (two decimal positions). Two
C files are created: NEWDAT contains all the depth measurements (x,y,
C z); BOTTYP contains the information on bottom characteristics.
C
C     Written by Karin Forney  12/87    Last modified 4/88
C=======================================================================
C Variables used:
C    code       i    cartographic code; determines format for depth.
C    cnt        i    keeps track of number records with depth values.
C    den        r    denominator when depth given in unit and fraction.
C    dep1       i    raw depth, (CODE determines how to convert this).
C    dep2       r    converted depth (into decimal of given unit).
C    dgr()      i    degrees latitude and longitude as read from file.
C    fillen     i    # of records in the data file to be read.
C    i          i    loop counter for loop converting each record.
C    mnt()      i    minutes of latitude and longitude as read from fil
C    num        r    numerator when depth given in unit and fraction.
C    sc1        i    integer representation of seconds of latitude and
C    sc2        i      longitude; used to write to BOTTYP file.
C    scn()      r    seconds of latitude and longitude as read from fil
C    ten        i    used to convert measurements into decimal units.
C    whol       i    whole portion of depth measurement.
C    x          r    contains decimal degrees latitude.
C    y          r    contains decimal degrees longitude.
C    z          r    contains depth converted into meters.
C
C=======================================================================
C
      program convert
C
C Initialization
C
      real dep2,ten,num,den,scn(2),x,y,z
      integer i,dep1,whol,sc1,sc2,cnt,fillen,code,dgr(2),mnt(2)
      cnt = 0
C
C This is for the CRAY; it also opens the necessary files
C
      call link('unit8=(datfil,open,text),unit9=(bottyp,create,text),
     1     unit10=(newdat,create,text)//')
C
C  Prompt for input from terminal to get length of the data file.
C
      write (6,*) 'How many records (lines) are in the data file?'
      read (5,*) fillen
C
C  Start reading the records and determine what needs to be done.
```

```
C
      do 100 i = 1,fillen
         read(8,10) dgr(1),mnt(1),scn(1),dgr(2),mnt(2),scn(2),dep1,code
  10     format(11x,i2,i2,f4.2,i3,i2,f4.2,i5,i3)
C
C  First case is if it is a bottom descriptor instead of a depth record
C  The info should be written to the file BOTTYP.
C
         if (((code.ge.533) .and. (code.le.599)) .or. (code .eq. 103)
     1    .or. (code .eq. 146)) then
            sc1 = int(scn(1)*100)
            sc2 = int(scn(2)*100)
            write(9,20) dgr(1),mnt(1),sc1,dgr(2),mnt(2),sc2,dep1,code
  20        format(11x,i2,i2,i4,i3,i2,i4,i5,i3)
            go to 100
C
C Second case is if depth is recorded in fathoms and feet (code=125).
C This is converted to decimal fathoms.
C
         elseif (code.eq.125) then
            whol = int(dep1/10)
            ten = (dep1-(whol*10.0))/6.0
            dep2 = whol + ten
C
C Third case is if depth is recorded in whole units. No conversion.
C
         elseif ((code.eq.126).or.(code.eq.129.).or.(code.eq.132)) then
            dep2 = real(dep1)
C
C Fourth case is if depth is recorded in units and tenths. Decimal
C point is placed in correct position.
C
         elseif ((code.eq.127).or.(code.eq.130).or.(code.eq.133)) then
            whol = int(dep1/10)
            ten =  (dep1 - (whol*10))/10.0
            dep2 = whol + ten
C
C Fifth case is if depth is recorded in units and fractions. Fraction
C is calculated and converted to decimal.
C
         elseif ((code.eq.128).or.(code.eq.131)) then
            whol = int(dep1/100)
            ten = int(dep1/10)
            num = ((ten*10.0) - (whol*100.0))/10.0
            den = dep1 - (ten*10.0)
            dep2 = whol + (num/den)
         end if
C
C  Now convert depth into meters if necessary (as determined by code)
C  Also add one to count of depth measurements found.
C
         if ((code.eq.125).or.((code.le.131).and.(code.ge.129))) then
            z = 1.8288*dep2
         elseif ((code.ge.126).and.(code.le.128)) then
            z = .3048*dep2
```

```
          else
             z = dep2
          end if
          cnt = cnt + 1
C
C  Now transform the x and y coordinates into decimal degrees and
C  print all three values to the NEWDAT file.
C
          y = dgr(1) + (mnt(1)/60.0) + (scn(1)/3600.0)
          x = dgr(2) + (mnt(2)/60.0) + (scn(2)/3600.0)
          x = -x
          write(10,50) x,y,z
   50     format(f11.6,2x,f11.6,2x,f7.2)
C
  100   continue

C
C Finally, let user know how many depth records there were in the file
C (this will determine the dimensions of the arrays which are passed
C to the contouring program CONRAN).
C
      write (6,*) '   '
      write (6,*) ' CONVERSION COMPLETED !'
      write (6,*) '   '
      write (6,*) 'The number of depth records in NEWDAT is: ' , CNT
      write (6,*) 'Please record this number; you will need it later.'
C
C End of program -- files close automatically.
C
      end
```

APPENDIX B

Program used to digitize depth points from navigation charts (written in Microsoft QuickBASIC on an IBM PC).

```
'-------------------------------------------------------------
' This program will accept points of latitude and longitude
' from a digitizer and append them to a specified data file.
' The information must be entered for a rectangular area,
' which is specified at the beginning of the program.
'
'  Written by Karin Forney                 January 88
'
'-------------------------------------------------------------
DIM x(2), y(2), xd(2), yd(2)
npts = 0

ON COM(1) GOSUB getlinput          'Process info when it comes
COM(1) ON                          'through serial port 1

OPEN "com1:9600,n,7,1,ds,rs,cs,cd" FOR INPUT AS #3

10 :
PRINT ""
PRINT "Lower left corner info:"
INPUT "Degrees and minutes longitude? (eg. 120,20) "; lngd, lngm
INPUT "Degrees and minutes latitude? (eg. 36,40)   "; latd, latm
       x(1) = lngd + (lngm / 60): y(1) = latd + (latm / 60)
PRINT ""
PRINT "Upper right corner info:"
INPUT "Degrees and minutes longitude? (eg. 120,20) "; lngd, lngm
INPUT "Degrees and minutes latitude? (eg. 36,40)   "; latd, latm
       x(2) = lngd + (lngm / 60): y(2) = latd + (latm / 60)
PRINT "coordinates: "; x(1); y(1), x(2); y(2)
PRINT ""
INPUT "Which file should this data be appended to"; nm$
OPEN nm$ FOR APPEND AS #1
PRINT ""
PRINT "Please start entering points with the digitizer."
PRINT "After setting origin near lower left corner, begin "
PRINT "with the lower left corner, then upper right corner,"
PRINT "and then enter all points with depth values."
PRINT ""

15 GOTO 15                         'This loop is terminated when
   STOP                            'the digitizing pen is set out
                                   'of bounds (see getlinput).
getlinput:
  i$ = INPUT$(10, #3)
  IF INSTR(i$, "9999") > 0 THEN STOP
  junk$ = INPUT$(2, #3)
  npts = npts + 1
  IF npts < 3 THEN
    xd(npts) = VAL(MID$(i$, 1, 5))
    yd(npts) = VAL(MID$(i$, 6, 5))
  ELSE
```

```
    nx = VAL(MID$(i$, 1, 5))
    ny = VAL(MID$(i$, 6, 5))
    realx = x(1) + (((nx - xd(1)) / (xd(2) - xd(1))) * (x(2) - x(1)))
    realy = y(1) + (((ny - yd(1)) / (yd(2) - yd(1))) * (y(2) - y(1)))
    PRINT ""
    PRINT nx; " = "; realx, ny; " = "; realy
    INPUT "DEPTH: ", d
    d = d * 1.8288
    PRINT #1, -realx, realy, d
  END IF
RETURN
```

APPENDIX C
Sample file with contouring instructions for program CONTOUR

```
title Plot for g35120
xlabel Degrees longitude
ylabel Degrees latitude
file g35120
read   100  100
level 0,10,20,30,40,50,60,70,80,90,100,
110,120,130,140,150,160,170,180,190,200
code 2 -1 -1 -1 -1 2 -1 -1 -1 -1 2 -1 -1 -1 -1 2 -1 -1 -1 -1 2
axes  -121.000     -120.000      35.0000      36.0000
height   7.00000
width    7.00000
plot
stop
```